

# 메타데이터를 이용한 임베디드 S/W 재사용 지원 시스템 설계

송문섭\*, 김철홍\*

\*한국전자통신연구원

## Design Embedded S/W Reuse Supporting System Using Metadata

Song, Moon Sub, Kim, Chul Hong

Electronics & Telecommunications Research Institute

E-mail : sirius@etri.re.kr, kch@etri.re.kr

### 요약

S/W공학에서는 오랫동안 S/W의 재사용에 대해 연구해 왔지만, H/W 분야에서의 재사용처럼 활성화되진 못하였다. 이러한 가장 큰 이유로는 S/W 개발의 초기부터 재사용을 염두에 둔 개발이 아닌 일회성 개발을 하기 때문이다. 즉 개발 시작 단계인 요구사항 수집에서부터 해당 도메인에 대한 지식을 바탕으로 분석한 후 설계하지 않기 때문이다. 이는 S/W 설계에 관한 전문 지식을 갖고 있는 개발자의 부재와 더불어 주어진 개발 기간이 짧은 국내 S/W 개발 업계의 현실이 열약하기 때문이다. 그럼에도 불구하고 많은 S/W 개발 업체들은 개발기간의 단축과 개발비용의 절감을 위해 기존 S/W를 재사용하고자 한다. 재사용하고자 하는 자산이 외부의 공개 S/W든 회사 자체의 기존 자산이든 이 때 발생하는 문제가 원하는 기존 자산을 어떻게 찾아 신규 개발에 활용할 수 있는가이다. 최악의 경우에는 필요한 자산을 찾아 변경하는 것이 신규로 개발하는 것보다 더 비효율적인 경우도 발생한다. 본 논문에서는 임베디드 S/W 분야에서 S/W 자산을 재사용하기 위해 필요한 지원 시스템으로 자산들을 메타데이터로 등록하고 검색할 수 있는 지원 도구를 설계하고자 한다.

### 1. 서론

S/W공학의 많은 개발 방법론에서는 체계화된 S/W 개발을 통해 제품의 생산성과 품질의 향상을 목표로 하고 있다.[1][2] 이러한 생산성과 품질의 향상은 최종 제품 생산 전에 산출되는 자산(모듈, 컴포넌트 등)의 개발과 관리에 따라 좌우된다.[3] 기존에 개발한 기업 내 자산 또는 오픈 소스에서 개발된 자산들을 이용하면 개발 기간을 단축시킬 수 있으며, 자산의 형상 관리를 통해 최종 제품에 대한 품질 향상을 꾀할 수도 있다.[4] S/W의 재사용을 위해서는 기본적으로 기업 내에 재사용에 대한 문화가 정착되어야 하며 이러한 문화가 활성화

되기 위해선 재사용 가능한 자산을 개발하기 위한 개발 도구와 재사용 자산을 검색, 획득 등을 지원해 주는 지원 도구가 필요하다.[5][6] 본 논문에서는 임베디드 S/W 개발 과정에서 산출되는 자산들을 재사용 가능한 형태의 메타데이터로 정의하여 자산 검색 및 활용이 용이한 재사용 지원 시스템을 설계하고자 한다. 이 때 고려해야 할 사항으로는 임베디드 S/W 도메인의 특성이 H/W에 종속되는 부분이 많으므로 이에 대한 정보를 재사용 자산 정보에 포함되어야 한다는 점이다. 또한 자산을 분야, 타입, 기능 등에 따라 분류할 수 있는 체계와 기준이 필요하다. 본 논문에서는 이러한 문제를 해결하기 위해 자료에 대한 국제 표준으로 자리

잡고 있는 더블린 코어 메타데이터를 이용하여 임베디드 S/W 재사용 자산을 정의하고 이를 구현하기 위한 시스템 설계를 하였다. 설계 문서 산출물은 컴포넌트 개발 방법론인 마르미III 개발 방법론의 산출물 양식에 따라 작성하였다.[7]

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 S/W 재사용과 더블린 코어에 대해 살펴보고 3장에서는 재사용 지원 시스템에 필요한 요구사항을 정의하겠다. 4장에서는 구현하고자 하는 재사용 지원 시스템을 설계하고 5장에서는 결론 및 향후 연구과제를 설명하겠다.

## 2. 관련 연구

### 2.1 S/W 재사용

S/W의 재사용은 소프트웨어공학에서뿐만 아니라 산업계에서도 많은 S/W 재사용 프로젝트를 통해 연구를 하고 있는 분야이다.[5][8] 그러나 S/W 재사용은 단기간에 효과를 볼 수 있는 기술이 아니며 최소 3년 이상 재사용 프로젝트를 수행해야 결과를 볼 수 있다. 성공적인 S/W 재사용 프로젝트들의 특징으로는 첫째 조직에서의 재사용에 대한 문화가 형성되어 있어야 한다. 둘째, 조직 구성원들에 대한 재사용 교육이 중요하다. 셋째, 적절한 인센티브로 재사용 동기 부여가 필요하다. 넷째, 사용하기 편한 개발 도구가 지원되어야 한다. 다섯째, 특정 분야에 대한 전문 지식을 보유하고 있어야 한다.[5][10] S/W 재사용에 필요한 세부 기술들로는 조직에 맞는 재사용 방법론, 재사용 자산에 대한 분류 기술, 재사용 자산의 개발을 지원하는 도구, 재사용 자산을 저장하고 관리하는 Repository 기술 등이 있다.[10][11]

### 2.2 더블린 코어

기존의 MARC 메타데이터는 구조의 경직성과 복잡성 때문에 자원을 기술하는데 많은 어려움이 있었다. 이를 대신하여 단순한 구조의 메타데이터 형식을 개발하여 자원의 신속한 검색 및 데이터의 호환성을 유지할 수 있는 방안을 모색하고자 OCLC와 NCSA가 1995년 3월 더블린에서 워크숍을 개최하여 합의한 기술이 더블린 코어이다. 더블린 코어는 메타데이터를 표현하는 표준 형식으로

16개의 기본 요소와 한정자(한정 요소와 인코딩)으로 이루어져 있다. 더블린 코어는 단순히 자원의 특성을 기술하는 것이 아니라 메타데이터 자체가 자원이 될 수 있다는 개념이며, 이러한 메타데이터를 만드는 가장 큰 목표는 원래 자원을 찾기 위한 중간 역할을 수행하기 위해서이다.[12]

## 3. 요구사항 명세

재사용 지원 시스템의 요구사항 명세 단계에 해당하는 마르미III 개발 방법론의 산출물로는 시스템 비전 기술서, 요구사항 기술서, 유스케이스 모형 기술서가 있다.(규모에 따른 마르미III 적용 시-수행기간 6개월 이하, 투입인력 20인 이하)

### 3.1 시스템 비전 기술

개발하고자 하는 임베디드 S/W 재사용 지원 시스템은 임베디드 S/W 재사용 자산의 보급, 확산을 위한 인프라를 개발자에게 제공함을 목적으로 하고 있다. S/W 재사용이 그룹 내에서 문화로 자리잡기 위해서는 이를 지원해 주는 시스템과 지원 환경 구축이 필수적이다. 임베디드 S/W 개발자가 개발한 재사용 자산을 정보저장소를 통해 공유함으로써 유사 기능의 중복 개발을 방지하고 새로운 시스템에 대한 개발 기간을 단축시킬 수 있다. 또한 재사용 자산에 대한 지속적인 버전 관리를 통해 재사용 자산의 품질을 향상시킬 수 있다. 다음 그림 1은 임베디드 S/W 재사용 지원 시스템의 개념도이다.

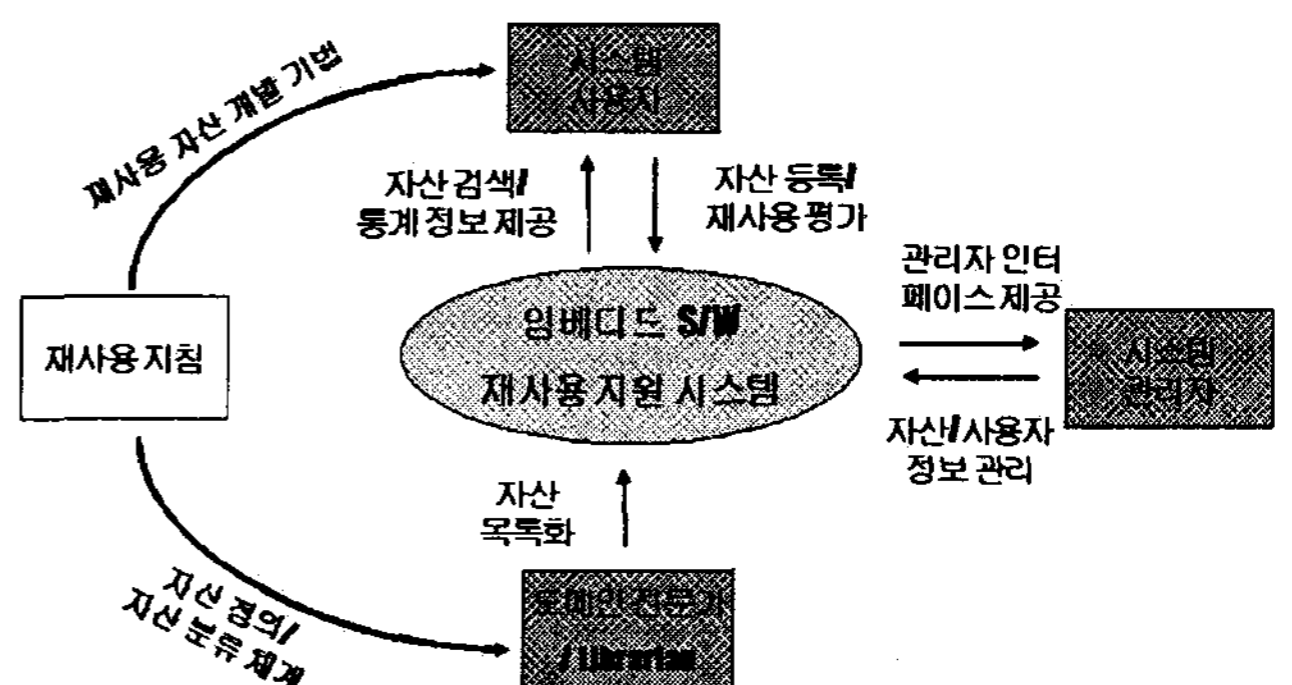


그림 1 임베디드 S/W 재사용 지원 시스템 개념도

### 3.2 요구사항 기술

요구사항을 기술하기 위해 먼저 시스템 사용자

를 정의하였다. 정의한 사용자로는 관리자, 사용자(소유자, 이용자), 도메인 전문가, 자산 분류 전문가, 품질 관리자 등이 있다. 다음으로는 16개의 상위 요구사항을 정의하였으며(R1-R16) 이를 바탕으로 45개의 세부 요구사항을 도출하였다.(DR001-DR045) 또한 이러한 요구사항을 기술하기 위해 필요한 7가지 비즈니스 규칙을 테이블로 정의하였다.(BR1-BR7)

### 3.3 유스케이스 모형 기술

유스케이스를 작성하기 위해 먼저 액터 목록을 정의하였다. 다음 그림 2는 임베디드 S/W 재사용 지원 시스템의 전체 유스케이스 다이어그램이다.

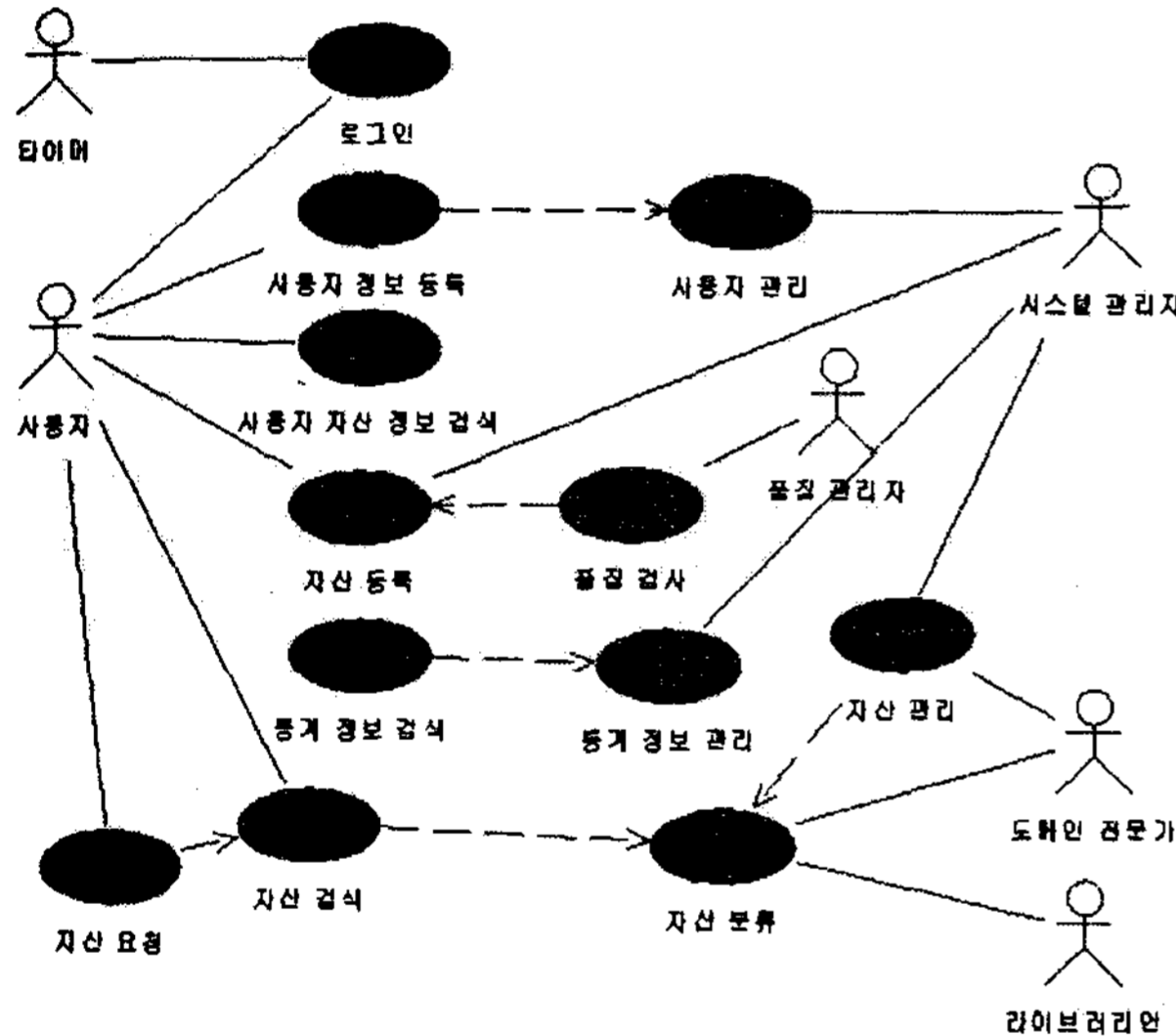


그림 2 임베디드 S/W 재사용 지원 시스템 전체 유스케이스 다이어그램

유스케이스 목록에는 유스케이스 이름, 관련 요구사항, 설명, 가변성 등을 속성으로 갖고 있다.(UC1-UC11) 유스케이스 설명에는 유스케이스 개요, 사건 흐름, 사전/사후 조건, 확장점, 관련 화면 등이 기술되어 있다.

이외에도 요구사항 명세서에는 데이터 사전과 데이터 모델, 분석 모델 등의 정보들이 기술되어 있다. 다음 그림 3은 자산 관련 데이터 모델이고 그림 4는 자산 등록 분석 모델의 예이다.

## 4. 재사용 지원 시스템 설계

### 4.1 개략 설계

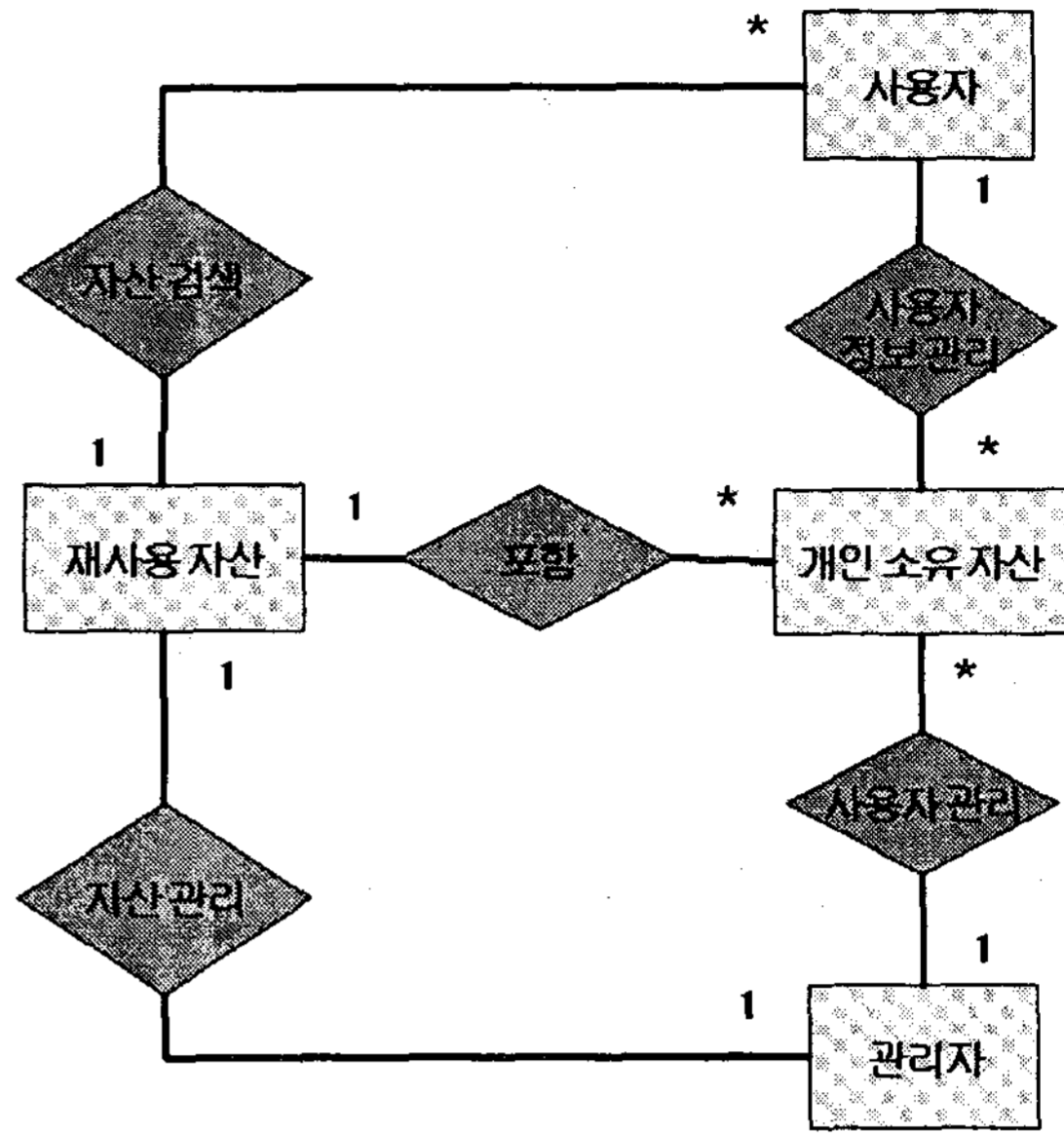


그림 3 자산 관련 데이터 모델

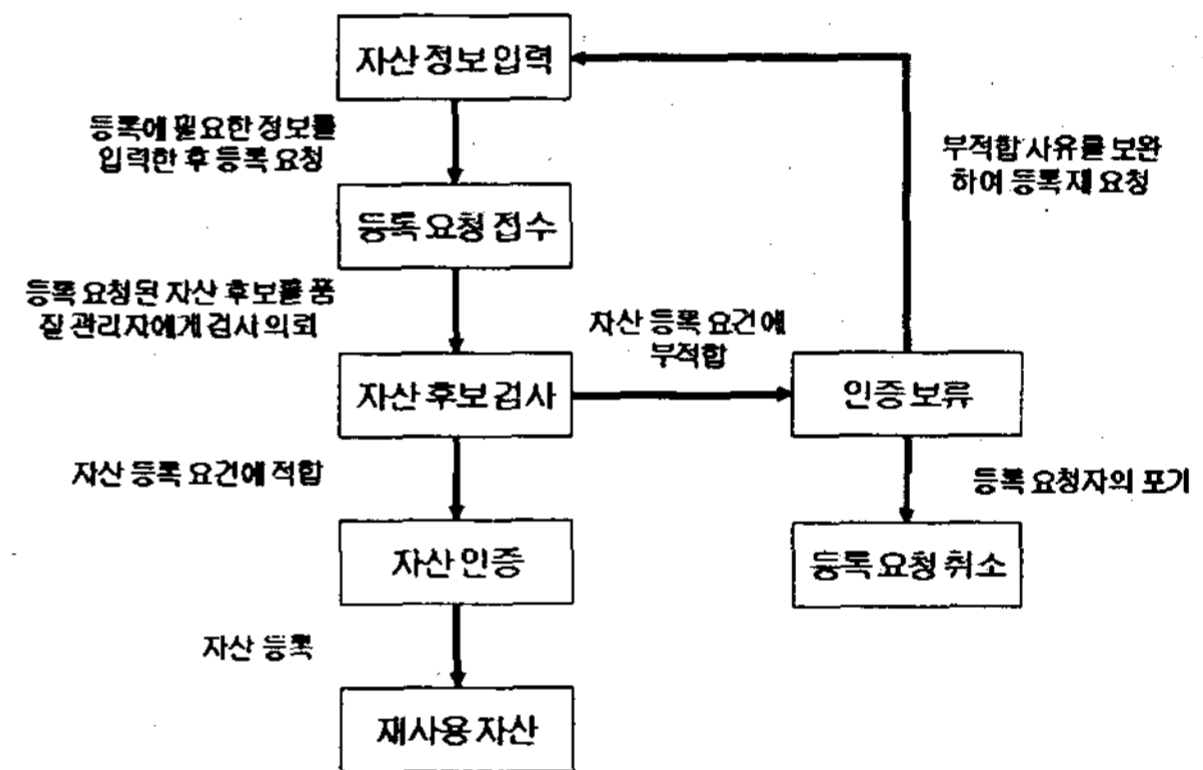


그림 4 자산 등록 분석 모델

개략 설계 단계에 포함되는 마르미III 산출물로는 시스템 아키텍처, 컴포넌트 명세, 객체 모형 등이 있다. 시스템 아키텍처 정의에서는 임베디드 S/W 재사용 지원 시스템의 물리적인 구조를 나타내며 다음 그림 5와 같다. 시스템에 접속하는 User는 크게 관리자와 사용자로 구분된다. 개발된 시스템은 기본적으로 개발 서버에 탑재된다. WebServer는 사용자가 웹을 통해 시스템에 접근할 수 있는 환경을 제공해 주며, Application Server는 컴포넌트를 배치시킴으로써 시스템이 제공하고자 하는 서비스를 수행한다. DB Server는 자료의 직접적인 관리를 담당한다. 이러한 Server들은 하나의 컴퓨터에 설치될 수도 있으며, 네트워크를 통해 연결되어 구축될 수도 있다.

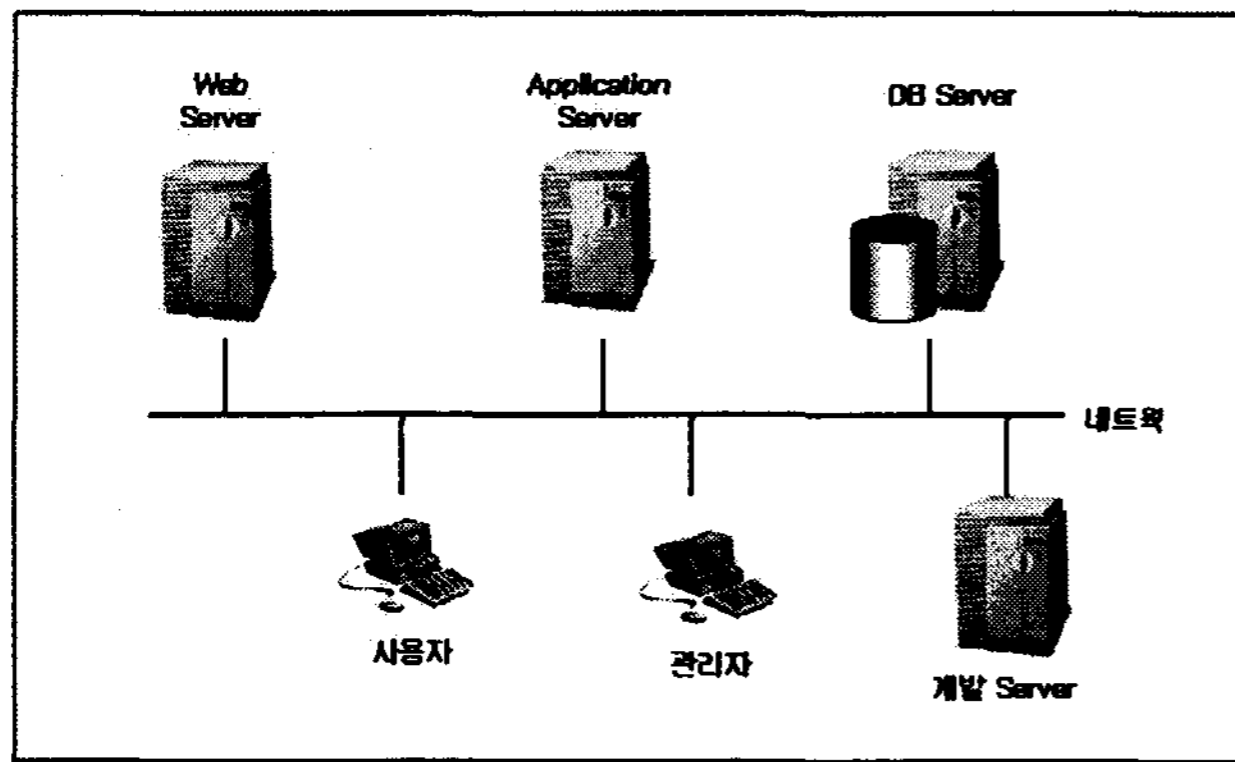


그림 5 시스템 아키텍처

구현 플랫폼을 적용한 시스템 아키텍처는 다음 그림 6과 같다.

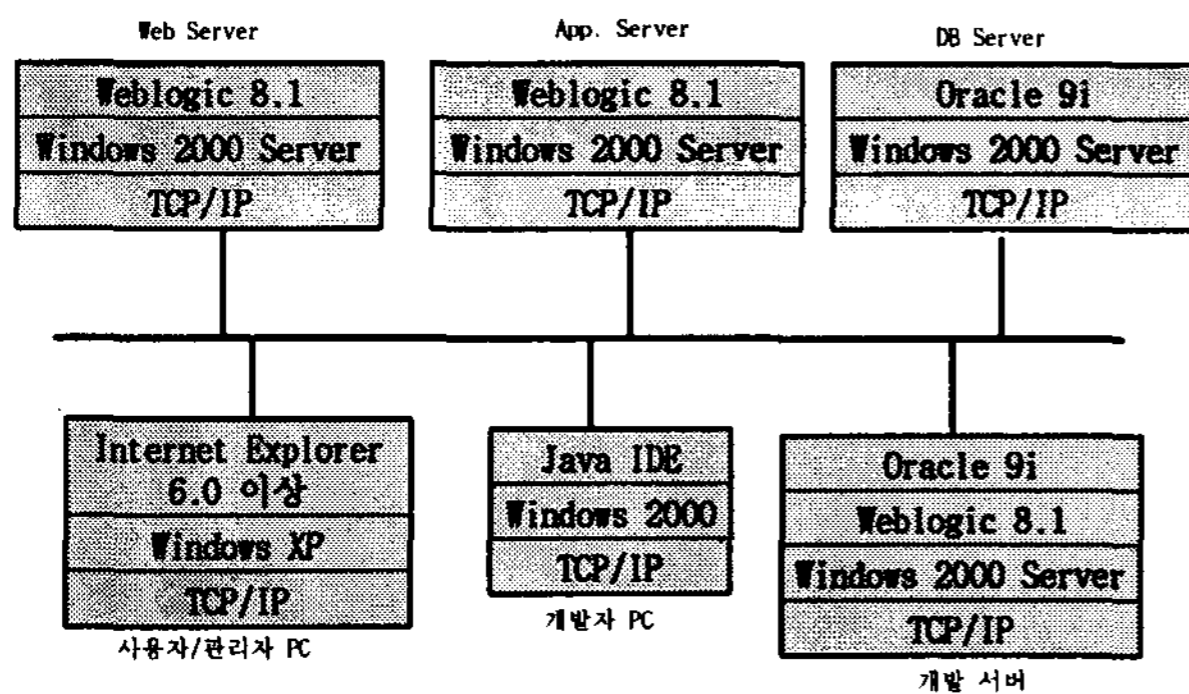


그림 6 구현 플랫폼을 적용한 시스템 아키텍처

소프트웨어 아키텍처는 소프트웨어 컴포넌트에 대해 패키지별 구성 형태를 나타낸다. UML 표현 기법을 이용하여 표현된 임베디드 S/W 재사용 지원 시스템의 소프트웨어 아키텍처는 다음 그림 7과 같다. 최상위 계층은 클라이언트 계층으로 익스플로어를 통해 서버 시스템에 접근하며 화면 계층은 이벤트 처리를 하는 RequestHandler와 화면 구성과 화면 UI 구현을 담당하는 웹 컴포넌트로 구성된다. 비즈니스 계층에는 각종 서비스를 담당하는 비즈니스 개체들로 구성되며, 시스템 자원 계층에는 각종 데이터들을 관리하는 DBMS가 있다.

컴포넌트 명세 과정에서는 시스템 컴포넌트와 비즈니스 컴포넌트로 나누어서 각각에 대해 컴포넌트도, 명세표, 상호계약 조건 등을 기술하였다. 다음 그림 8은 임베디드 S/W 재사용 지원 시스템의 전체 컴포넌트도이다.

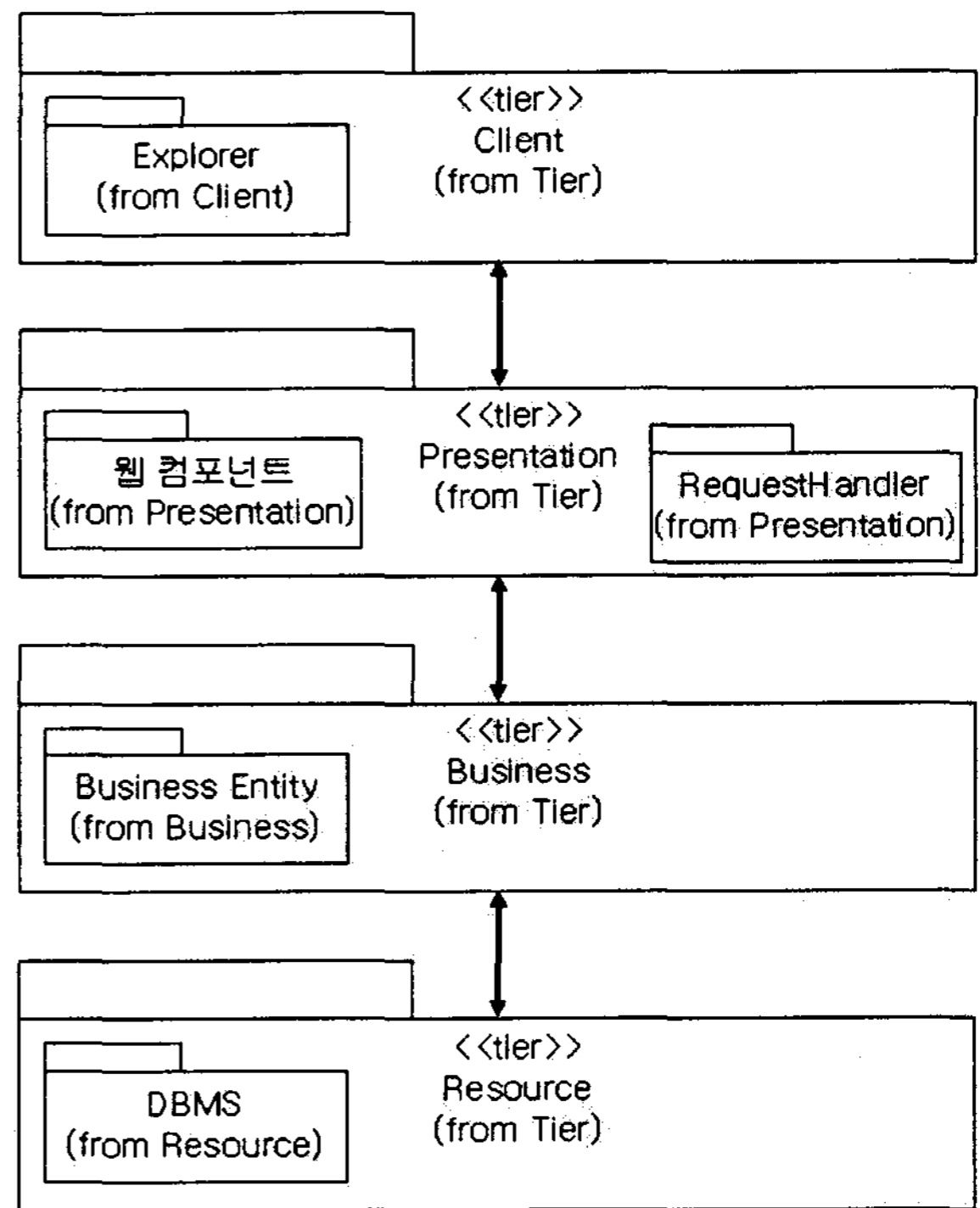


그림 7 소프트웨어 아키텍처

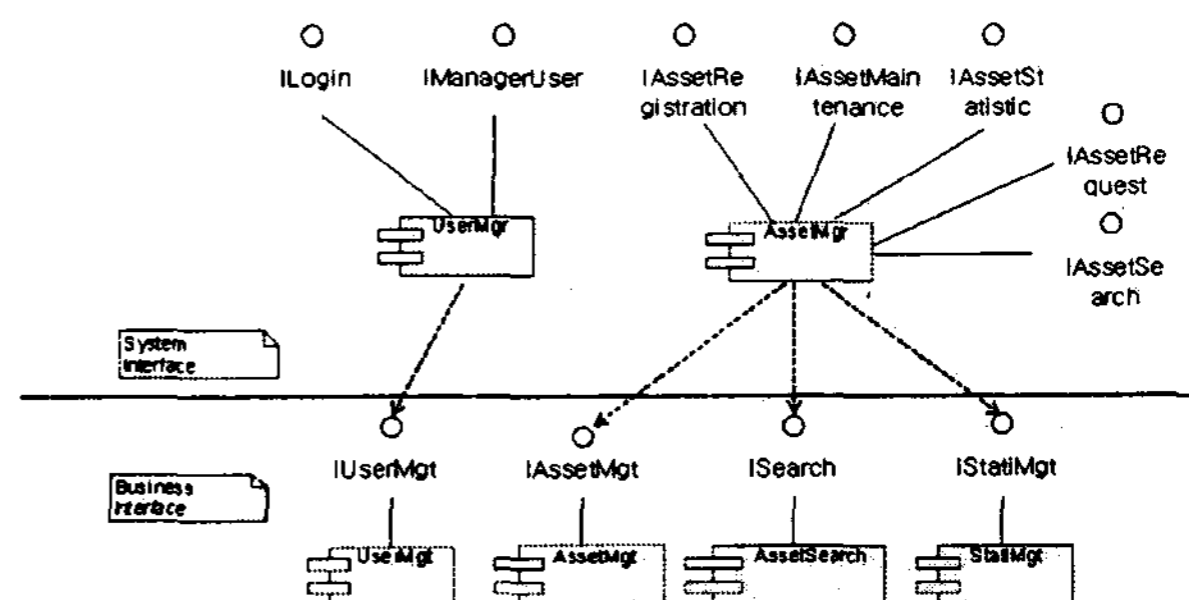


그림 8 임베디드 S/W 재사용 지원 시스템 컴포넌트도

객체 모형 기술 과정에서는 앞 단계에서 작성한 유스케이스 모형 별로 클래스 다이어그램을 기술하였다. 다음 그림 9는 임베디드 S/W 재사용 지원 시스템의 전체 클래스도이다.

#### 4.2 상세 설계

상세 설계 단계에서는 패키지 정의, EJB 정의, UI 상세 설계, DB 설계를 하였다.

패키지 정의는 3개의 서브 시스템, 5개의 컴포넌트 그리고 주요 클래스들에 대해 표로 기술하였다. 또한 각각의 주요 클래스들에 대해 구현 클래스 모형을 기술하였다.



- [5] Joos, R., Software Reuse at Motorola, *IEEE Software*, 42-47, 1994
- [6] Biddle, R.L, Tempero, E.D., "Towards tool support for reuse", *Software Engineering: Education and Practice, Proceedings*, pp.126-133, 1998
- [7] "컴포넌트 개발방법론 개발"(MARM I III), 한국전자통신연구원, 2003
- [8] Applied Expertise, Inc., Software Reuse Benchmarking Study: Learning from Industry and Government Leaders, Report for the US Department of Defense Software Reuse Initiative, , 1996
- [9] Card, D., and E. Comer, "Why do so Many Reuse Programs Fail?", *IEEE Software*, pp. 114-115, 1994
- [10] Weizhi Wang, Minchen Zhu, Bingham Liu, "A reusable software development method for intelligent control", WCICA 2004. Fifth World Congress on Volume 4, pp. 3260 - 3264, 2004
- [11] Torkar, R, Mankefors, S., "A survey on testing and reuse", *SwSTE'03. Proceedings. IEEE International Conference*, pp164 - 173, 2003
- [12] <http://dublincore.org>.